remark: For any 2 words $x, y \in \{0,1\}^*$, we denote their concatenation by: $x \circ y$. We'll also write $x \circ 0$, $x \circ 0^n$, and so.

Exe 1 one way functions and P **vs.** NP **(10 points).** Prove that the existence of one-way functions implies P $\neq$ NP.
Guideline: for any poly-time computable function f define a set $L_f \in$ NP such that if $L_f \in$ P then f in insertable (by poly-time algorithm)

solution 1: Assume otherwise, that is: P = NP. WLG, assume f is a length preserving OWF. We'll show that there is an efficient algorithm D, that can invert f.
Define the following language:

$$L_f = \{ < x, y > : \ x, y \in \{0,1\}^\star \ , |x| \leq |y| \quad \text{and} \quad \exists t \in \{0,1\}^\star \text{ s.t. } f(x \circ t) = y\}$$

So, L_f contains all pairs of words $< x, y >$ such that x can be extended to a word w such that $f(w) = y$.
Clearly $L_f \in NP$. This is because we can use t as the witness, and applying f to the string $x \circ t$ can be done in polynomial time. Since we assumed that $P = NP$, we conclude that there is a deterministic algorithm A that can decide whether the pair of words $< x, y >$ belongs to L_f. Denote a call to A, by $A(x, y)$.
We'll describe now, a poly time algorithm D that can invert our OWF, f:

Algorithm 0.1 (invert f)**.**

input: 1^n , $y = f(x) \in \{0,1\}^*$

- $x \leftarrow \epsilon$ *(the empty string)*
- *while* $(|x| < |y|)$
- *if* $(A(x \circ 0, y) ==$ *true)*
- $x \leftarrow x \circ 0$
- *else*
- $x \leftarrow x \circ 1$
- *return* x;

Clearly, the above algorithm is a poly time, and it inverts $f(x)$ for every x. That of course contradict the fact that f is OWF.

Exe 2 (10 points). Refute the following conjecture:

For every length-preserving one-way function f, the function $f'(x) = f(x) \oplus x$ is one-way.

solution 2: Suppose g is a OWF, length preserving. Define a function f as:

$f : \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ $\quad\quad f(x) = 0^n \circ g(x_{1\dots n})$ $\quad\quad (x \text{ of length } 2n)$

Claim 0.2. *f is length preserving OWF*

Proof of Claim 0.2. It's obvious that f is length preserving. Assume f isn't a OWF. So There exist a PPT algorithm A, a poly $p(n)$ and an infinite set $I \subseteq \{2k : k \in N\}$ such that for every $n \in I$:

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(1^n, f(x)) \in f^{-1}(f(x))] > \frac{1}{p(n)}$$

We'll build an algorithm B that will invert g:

Algorithm 0.3 (B - invert g)**.**

<u>input</u>*:* 1^n , $y = g(x) \in \{0,1\}^n$

<u>output</u>*:* $A(1^{2n}, 0^n \circ y)_{1\dots n}$

The following holds:

$$\begin{aligned}
\mathsf{Pr}_{x \leftarrow \{0,1\}^n}[B(1^n, g(x)) \in g^{-1}(g(x))] &= \mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(1^{2n}, 0^n \circ g(x))_{1\dots n} \in g^{-1}(g(x))] \\
&= \mathsf{Pr}_{x \leftarrow \{0,1\}^n, w \leftarrow \{0,1\}^n}[A(1^{2n}, f(x \circ w))_{1\dots n} \in g^{-1}(g(x))] \\
&= \mathsf{Pr}_{x \leftarrow \{0,1\}^n, w \leftarrow \{0,1\}^n}[g(A(1^{2n}, f(x \circ w))_{1\dots n}) = g(x)] \\
&= \mathsf{Pr}_{x \leftarrow \{0,1\}^n, w \leftarrow \{0,1\}^n}[0^n \circ g(A(1^{2n}, f(x \circ w))_{1\dots n}) = 0^n \circ g(x)] \\
&= \mathsf{Pr}_{x \leftarrow \{0,1\}^n, w \leftarrow \{0,1\}^n}[f(A(1^{2n}, f(x \circ w)) = f(x \circ w)] \\
&= \mathsf{Pr}_{x \leftarrow \{0,1\}^{2n}}[f(A(1^{2n}, f(x)) = f(x)] \geq \frac{1}{p(2n)}
\end{aligned}$$

Of course, that contradict the hardness of g. □

Remark 0.4. *As we saw in class, f can easily be extended to be a OWF of any length (not just even).*

Claim 0.5. *$f'(x) = x \oplus f(x)$ isn't OWF*

Proof of Claim 0.5. We'll see how to build a PPT algorithm that will invert f', for any input of even length. First we notice that for every $x \in \{0,1\}^{2n}$ we have:

$$\begin{aligned} f'(x) &= x \oplus f(x) \\ &= (x_{1\dots n} \circ x_{n+1\dots 2n}) \oplus (0^n \circ g(x_{1\dots n})) \\ &= (x_{1\dots n} \oplus 0^n) \circ (x_{n+1\dots 2n} \oplus g(x_{1\dots n})) \\ &= (x_{1\dots n}) \circ (x_{n+1\dots 2n} \oplus g(x_{1\dots n})) \end{aligned}$$

So, the first n bits of $f'(x)$ are actually $x_{1\dots n}$. To get $x_{n+1\dots 2n}$ we notice that:

$$x_{n+1\dots 2n} \oplus g(x_{1\dots n}) \oplus g(x_{1\dots n}) = x_{n+1\dots 2n}$$

Hence the following PPT algorithm will invert f' for every even input:

Algorithm 0.6 (invert f')**.**

input: 1^k , $y = f'(x) \in \{0,1\}^k$

- *if k is odd*
- *do whatever you want*
- *else*
- *return:* $y_{1\dots n} \circ (y_{n+1\dots 2n} \oplus g(y_{1\dots n}))$ *(where $n = k/2$)*

$\square$

Exe 3 (10 points). Let f be a one-way function. Prove that for any PPT A, it holds that

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n, i\leftarrow[n]}\left[\mathsf{A}(f(x),i) = x[i]\right] \leq 1 - \frac{1}{2n},$$

for large enough $n \in \mathbb{N}$, where $x[i]$ is the i'th bit of x.
Bonus* : prove the above when replacing $1-\frac{1}{2n}$ with $1-\frac{1}{n}$.

Solution 3: Assume by contradiction that there in a PPT algorithm A, and infinitely many $n \in N$, such that:

$$Pr_{x\leftarrow\{0,1\}^n, i\leftarrow[n]}[A(f(x),i) = x[i]] > 1 - \frac{1}{2n}$$

We'll build a PPT algorithm B that invert f.
Algorithm B:
input: 1^n, $y = f(x)$
for every position $i \in [1 \dots n]$, let $t[i] = A(f(x), i)$;
return t;

Choose some random $x \in \{0,1\}^n$ and run B on $f(x)$. What is the probability that B fails on $f(x)$?

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n}[(B(f(x)) \neq x] = \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[\bigvee_{t=1}^{n} A(f(x),t) \neq x[t]]$$

$$\leq \sum_{t=1}^{n} \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[A(f(x),t) \neq x[t]] \tag{1}$$

Now lets evaluate the last sum. Using conditional probability we have:

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n, i\leftarrow[n]}[A(f(x),i) \neq x[i]] \ = \sum_{t=1}^{n} \mathsf{Pr}[i=t] \cdot \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[A(f(x),t) \neq x[t]] =$$

$$= \frac{1}{n} \cdot \sum_{t=1}^{n} \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[A(f(x),t) \neq x[t]]$$

Since:

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n, i\leftarrow[n]}[A(f(x),i) \neq x[i]] < \frac{1}{2n}$$

We conclude:

$$\sum_{t=1}^{n} \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[A(f(x),t) \neq x[t]] < \frac{1}{2}$$

Going back to (1) we have:

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n}[(B(f(x)) \neq x] < \frac{1}{2}$$

Hence:

$$\mathsf{Pr}_{x\leftarrow\{0,1\}^n}[(B(f(x)) = x] > \frac{1}{2}$$

Which for sure break the hardness of f.

bonus: Declare an algorithm C as follow:

Algorithm 0.7.

input: 1^n, $y = f(x)$

- *Choose* $pos \leftarrow \{1, 2, \ldots n\}$
- *for* $(i = 1$ *to* $n)$ *do*
- *if* $(i \neq pos)$
- $x'[i] \leftarrow A(f(x), i)$
- *Choose* $x'[pos] \leftarrow \{0, 1\}$;
- *if*$(f(x') == y)$
- *return* x';
- *else*
- $x'[pos] = 1 - x'[pos]$;
- *return* x';

So except of a random position pos, C act the same as B. For that random position, C tries both options of 0/1 in order to succeed.
What is the probability for C to fail on $f(x)$? It needs to fail on at least one of the bits, except pos. On pos it can never fail since both options are checked. We have:

$$\begin{aligned}
\mathsf{Pr}_{x\leftarrow\{0,1\}^n}[C(f(x)) \neq x] &= \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[\bigvee_{t=1}^{n} C(f(x))[t] \neq x[t]] \\
&\leq \sum_{t=1}^{n} \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[C(f(x))[t] \neq x[t]] \\
&= \sum_{t=1}^{n} \mathsf{Pr}[t = pos] \cdot \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[C(f(x))[t] \neq x[t] \mid t = pos] \\
&+ \sum_{t=1}^{n} \mathsf{Pr}[t \neq pos] \cdot \mathsf{Pr}_{x\leftarrow\{0,1\}^n}[C(f(x))[t] \neq x[t] \mid t \neq pos]
\end{aligned}$$

The first sum is 0. Continue developing the second sum, with the observation that $t \neq pos \Rightarrow C(f(x))[t] = A(f(x), t)$:

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^n}[C(f(x)) \neq x] \leq \frac{n-1}{n} \cdot \sum_{t=1}^{n} \mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(f(x), t) \neq x[t]] \tag{2}$$

As before using conditional probability we have:

$$\begin{aligned}
\mathsf{Pr}_{x \leftarrow \{0,1\}^n, i \leftarrow [n]}[A(f(x), i) \neq x[i]] &= \sum_{t=1}^{n} \mathsf{Pr}[i = t] \cdot \mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(f(x), t) \neq x[t]] \\
&= \frac{1}{n} \cdot \sum_{t=1}^{t \leq n} \mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(f(x), t) \neq x[t]]
\end{aligned}$$

Since:

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^n, i \leftarrow [n]}[A(f(x), i) \neq x[i]] < \frac{1}{n}$$

We conclude:

$$\sum_{t=1}^{t \leq n} \mathsf{Pr}_{x \leftarrow \{0,1\}^n}[A(f(x), t) \neq x[t]] < 1$$

Substituting it in (2):

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^n}[C(f(x)) \neq x] < \frac{n-1}{n}$$

Hence:

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^n}[C(f(x)) = x] > \frac{1}{n}$$

Which contradict the hardness of f.

Exe 4 (basic probability). Let P and Q be distributions over a finite set $\mathcal{U}$.

a. (2 points) Prove that $\mathrm{SD}(P,Q) = \max_{\mathcal{S}\subseteq\mathcal{U}}(P(\mathcal{S}) - Q(\mathcal{S}))$ (recall that $\mathrm{SD}(P,Q) := \frac{1}{2}\sum_{u\in\mathcal{U}} |P(u) - Q(u)|)$).

b. (3 points) Prove that $\mathrm{SD}(P^2, Q^2) \leq 2 \cdot \mathrm{SD}(P,Q)$ (see "Notation" in the first class slides for the definition of P^2, Q^2).

Let $\mathcal{Q} = \{Q_n\}_{n\in\mathbb{N}}$, $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ and $\mathcal{R} = \{R_n\}_{n\in\mathbb{N}}$ be distribution ensembles.

c. (2 points) Given that $\mathcal{Q} \overset{c}{\equiv} \mathcal{P}$ (i.e., $\mathcal{Q}$ is computationally indistinguishable from $\mathcal{P}$) and $\mathcal{P} \overset{c}{\equiv} \mathcal{R}$, prove that $\mathcal{Q} \overset{c}{\equiv} \mathcal{R}$.

d. (3 points) Give an example for ensemble $\mathcal{Q}$ and $\mathcal{P}$ such that: (1) $\mathrm{Supp}(Q_n) = \mathrm{Supp}(P_n)$ for every $n \in \mathbb{N}$, and (2) $\mathrm{SD}(Q_n, P_n) = 1 - \mathrm{neg}(n)$ (i.e., for every $p \in \mathrm{poly}$, exists $n' \in \mathbb{N}$ with $\mathrm{SD}(Q_n, P_n) > 1 - \frac{1}{p(n)}$ for every $n > n'$)

Solution 4(a): Denote the following:

$U_{P>Q} := \{u \in U \mid P(u) > Q(u)\}$
$U_{Q>P} := \{u \in U \mid Q(u) > P(u)\}$
$U_{Q=P} := \{u \in U \mid Q(u) = P(u)\}$

Obviously we have:

$$P(U_{P>Q}) + P(U_{Q>P}) + P(U_{P=Q}) \;=\; Q(U_{P>Q}) + Q(U_{Q>P}) + Q(U_{P=Q}) \;=\; 1$$

Since $P(U_{P=Q}) = Q(U_{P=Q})$, We get:

$$P(U_{P>Q}) - Q(U_{P>Q}) \;=\; Q(U_{Q>P}) - P(U_{Q>P}) \tag{3}$$

The following holds:

$$\begin{aligned}
\frac{1}{2}\sum_{u\in U} |P(u) - Q(u)| &= \frac{1}{2}\sum_{u\in U_{P>Q}} |P(u) - Q(u)| + \frac{1}{2}\sum_{u\in U_{Q>P}} |P(u) - Q(u)| \\
&= \frac{1}{2}\sum_{u\in U_{P>Q}} P(u) - Q(u) + \frac{1}{2}\sum_{u\in U_{Q>P}} Q(u) - P(u) \\
&= \frac{1}{2}\cdot (P(U_{P>Q}) - Q(U_{P>Q})) + \frac{1}{2}\cdot (Q(U_{Q>P}) - P(U_{Q>P})) \\
&= P(U_{P>Q}) - Q(U_{P>Q})
\end{aligned}$$

The last equally is due to (3).

We also note that:

$$\max_{S\subseteq U}(P(S) - Q(S)) = P(U_{P>Q}) - Q(U_{P>Q})$$

since adding an element u such that $P(u) = Q(u)$ won't change $P(U_{P>Q}) - Q(U_{P>Q})$, and adding u such that $P(u) < Q(u)$ will decrease it.

Solution 4(b):

$$
\begin{aligned}
SD(P^2, Q^2) &= \frac{1}{2} \cdot \sum_{(x,y)\in U^2} |P(x)P(y) - Q(x)Q(y)| \\
&= \frac{1}{2} \cdot \sum_{(x,y)\in U^2} |P(x)P(y) - P(x)Q(y) + P(x)Q(y) - Q(x)Q(y)| \\
&= \frac{1}{2} \cdot \sum_{(x,y)\in U^2} |P(x)(P(y) - Q(y)) + Q(y)(P(x) - Q(x))| \\
&\leq \frac{1}{2} \cdot \sum_{(x,y)\in U^2} P(x)\,|P(y) - Q(y)| + \frac{1}{2} \cdot \sum_{(x,y)\in U^2} Q(y)\,|P(x) - Q(x)| \\
&= \frac{1}{2} \cdot \sum_{x\in U} P(x) \cdot \sum_{y\in U} |P(y) - Q(y)| + \frac{1}{2} \cdot \sum_{y\in U} Q(y) \cdot \sum_{x\in U} |P(x) - Q(x)| \\
&= \frac{1}{2} \cdot 1 \cdot \sum_{y\in U} |P(y) - Q(y)| + \frac{1}{2} \cdot 1 \cdot \sum_{x\in U} |P(x) - Q(x)| \\
&= SD(P,Q) + SD(P,Q) = 2 \cdot SD(P,Q)
\end{aligned}
$$

solution 4(c): Assume on the contrary that $\mathcal{Q} \overset{c}{\not\equiv} \mathcal{R}$. So we have a PPT algorithm D, a poly $p(n)$, and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that for every $n \in \mathcal{I}$ we have:

$$
|\mathsf{Pr}_{x\leftarrow Q_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow R_n}[D(x) = 1]| > \frac{1}{p(n)}
$$

So:

$$
\frac{1}{p(n)} < |\mathsf{Pr}_{x\leftarrow Q_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow R_n}[D(x) = 1]| =
$$
$$
|\mathsf{Pr}_{x\leftarrow Q_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1] + \mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow R_n}[D(x) = 1]| \leq
$$
$$
|\mathsf{Pr}_{x\leftarrow Q_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1]| + |\mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow R_n}[D(x) = 1]|
$$

So for every $n \in \mathcal{I}$ we have either:

$$
|\mathsf{Pr}_{x\leftarrow Q_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1]| \geq \frac{1}{2p(n)} \tag{4}
$$

Or:

$$
|\mathsf{Pr}_{x\leftarrow P_n}[D(x) = 1] - \mathsf{Pr}_{x\leftarrow R_n}[D(x) = 1]| \geq \frac{1}{2p(n)} \tag{5}
$$

Since $\mathcal{I}$ is infinite set we must have one of: (4) hold for infinitely many n or (5) hold for infinitely many n. So either $\mathcal{Q} \overset{c}{\not\equiv} \mathcal{P}$ or $\mathcal{P} \overset{c}{\not\equiv} \mathcal{R}$, which contradict the assumptions.

solution 4(d): Lets take $\Omega = \{0, 1\}$. P_n and Q_n are defined as:

$P_n(0) = \frac{1}{2^n}$, $P_n(1) = 1 - \frac{1}{2^n}$

$Q_n(0) = 1 - \frac{1}{2^n}$, $Q_n(1) = \frac{1}{2^n}$

Definitely $\mathrm{Supp}(Q_n) = \mathrm{Supp}(P_n) = \{0, 1\}$

Also for every n we have:

$$\begin{aligned}
\mathrm{SD}(Q_n, P_n) &= \frac{1}{2} \sum_{u \in \{0,1\}} |P_n(u) - Q_n(u)| \\
&= \frac{1}{2} \cdot (|P_n(0) - Q_n(0)| + |P_n(1) - Q_n(1)|) \\
&= 1 - \frac{1}{2^{n-1}} \\
&= 1 - \mathrm{neg}(n)
\end{aligned}$$